前言：

Spring加载和注册Bean的基础类是DefaultListableBeanFactory，而通过xml文件的方式进行加载的基础类是XmlBeanFactory，其与DefaultListableBeanFactory的主要区别是，XmlBeanFactory类中自定义了自己的XmlBeanDifinitionReader。

详解BeanFactory bf = new XmlBeanFactory(new ClassPathResouce(“beanFactoryTest.xml));代码的详细过程

1. 将xml文件转换成ClassPathResource对象。不同来源的Resource有不同的实现类，FileSystemResource
2. 当对xml文件读取全部封装到Resource后，接下来的操作就主要在XmlBeanDefinitionReader类中了，主要的方法是loadBeanDefinitions(resource)方法。
3. 在进行调用loadBeanDefinitions()方法之前，调用了ignoreDependencyInterface(BeanNameAware.class)方法，这个步骤不是很清楚。
4. 深入的加载步骤，在loadBeanDefinations(rsource)中，首先将resource转换成EncodedResource(resource）对象，这个步骤主要是用于设置输入流的编码方式，然后在调用loadBeanDefinitions(EncodedResource encodedResource);在这个方法中将资源保存在resourcesCurrentlyBeingLoaded set集合中。然后获取到resource的inputStream和生成InputSource对象，调用doLoadBeanDefinitions(inputSource,encodedResource.getResource());
5. DoLoadBeanDefinitions()方法中，首先进行判断是采用dtd方式还是xsd方式来验证xml文件，获取到resource的inputStream对象，来获取到xml中的文件，并逐行读取，如果一行中包含doctype字符串，就采用dtd方式验证，若没有则采用xsd方式验证。
6. 获取到验证方式后，就准备生成Document对象了，在生成document对象时候，有EntityResolver对象传入，该对象时用于通过本地的xsd规范或者xsd规范来进行验证xml文件，而不需要通过网络下载xsd或dtd规范来验证。
7. LoadDocument（）方法返回Document对象，首先通过验证模式和命名空间获取到DocmentBuilderFactory对象，然后通过该对象，entityResolver和errorHandler来获取builder，然后通过builder.parse(inputSource);来获取的Document对象
8. 获取到document对象后，接下来就是解析和注册了，解析和注册的真正对象是BeanDefinitionDocumentReader, 通过其registerBeanDefinitions(doc,createReaderContext(resource));方法来进行解析
9. DoRegisterBeanDefinitions(Element root){}解析的核心方法，在真正解析过程中，首先判断是否有profile属性，如果有profile，则分别对profile进行操作。

如果xml的命名空间与spring的命名空间相同，则spring会使用默认的解析方式，如果是自定义的，也就是说命名空间不等于http://www.springframework.org/schema/beans则会使用自定义的解析方法

1. 默认标签一共有import、alias、bean、beans4种
2. bean标签解析过程：解析对象有id、name、parent、元数据、lookup-method、replaced-method、构造函数、property、qualifier子元素等。
3. import、alias、beans加载过程：<https://www.jianshu.com/p/d2859d991d6b>

<https://www.jianshu.com/p/e7c1d98b954d>

总结：xml加载bean的思想，设计模式：在创建DocumentBuilder对象，就使用了命令模式+工厂模式；面向接口编程：resource类；单一职责原则：一个方法就实现一个功能；封装：EncodedResource；面向继承：在真正解析xml时，有解析前方法和解析后方法，但这两个方法都是空的，这方便了如果继承了默认实现解析xml的类，重写了这两个方法，就能够达到自己想要的操作了。缺点：当判断当前节点的默认命名空间时候，直接通过写死的url来判断，感觉不妥。

<https://cloud.tencent.com/developer/article/1176597>